Amendments to the Claims:

Please amend claims 3-6, 8, 11, 17, 22, 24-26, 33, 34, 39-41, 54, 55, 57, 58, 61, 63, 66, 67, 70, and 78, and please cancel claims 1, 2, 57, 59, and 62. Please add new claims 79-82. Following is a complete listing of the claims pending in the application, as amended:

1. (Cancelled)

2. (Cancelled)

3. (Currently Amended) The method of claim ~~2~~79, wherein the authorization data is at least a biometric information sample, and wherein the authenticating includes comparing the at least a biometric information sample to a previously stored biometric template.

4. (Currently Amended) The method of claim ~~2~~79, further comprising:

when the comparison is indicative of other than unauthorized executable code resident in a computer system, providing the requested security data relating to the user.

5. (Currently Amended) The method of claim ~~1~~79, further comprising:

receiving a request for security data from an application in execution in the computer system; and

when the comparison is indicative of other than an unauthorized executable program resident in a computer system, providing security data to the application.

6. (Currently Amended) The method of claim ~~1~~79, wherein the trusted hash value and the computed hash value are determined by a same trusted security

application executing locally on a processor of a same computer system at different times, the trusted hash value determined when the known state is a secure state.

7. (Previously Presented) The method of claim 6, wherein the trusted hash value is digitally signed.

8. (Currently Amended) The method of claim 7, ~~wherein the hashing data includes:~~ further comprising verifying an authenticity of the digitally signed trusted hash value.

9. (Previously Presented) The method of claim 8, further comprising:

receiving a request for security data from an application in execution in the computer system; and

when the authenticity of the digitally signed trusted hash value is verified and the comparison is indicative of other than unauthorized executable code resident in a computer system, providing the requested security data to the application.

10. (Previously Presented) The method of claim 9, wherein the application and the predetermined hashing process are both executed on a same processor of the computer system.

11. (Currently Amended) The method of claim 8, further comprising:

when the computed hash value and the trusted hash value are other than indicative of a secure state, issuing a notification that unauthorized executable code is detected within the computer system.

12. (Previously Presented) The method of claim 11, further comprising:

when the computed hash value and the trusted hash value are other than indicative of a secure state, preventing access to the computer system.

13. (Previously Presented) The method of claim 7, further comprising:

transmitting the trusted hash value to a second other computer system in communication with the computer system and retrievably storing the computed hash value within the second other computer system.

14. (Previously Presented) The method of claim 13, further comprising transmitting the computed hash value to the second other computer system for comparison with the trusted hash value by a processor of the second other computer system.

15. (Previously Presented) The method of claim 14, wherein the computed hash value is determined in dependence upon the predetermined data existing in memory within the computer system and some time dependent data of the computer system.

16. (Previously Presented) The method of claim 14, wherein the second other computer system includes a trusted source and wherein security data is stored for provision to applications in execution on systems that are known to be secure.

17. (Currently Amended) A method of detecting unauthorized executable code resident in a computer system, the method comprising:

receiving user authorization information;

authenticating the user authorization information to perform at least one of authorize and identify a user;

when the user is at least one of authorized or identified, requesting security data of the user;

providing a trusted security application executable on a processor of the computer system for determining a hash value using a selected hashing process applied to predetermined data existing in memory within the

computer system, wherein the predetermined data includes system memory locations indicative of executable programs in operation;

hashing the selected data existing in memory within the computer system using the predetermined process to determine a hash value;

digitally signing the hash value to provide a trusted hash value; and

retrievably storing the trusted hash value, wherein the predetermined data relates to programs in execution on the processor of the computer system when the computer system is in a secure state.

18. (Previously Presented) The method of claim 17, further comprising:

comparing a computed hash value with the trusted hash value to detect changes to the predetermined data existing in memory within the computer system.

19. (Previously Presented) The method of claim 18, further comprising verifying the authenticity of the digital signature of the trusted hash value.

20. (Previously Presented) The method of claim 19, further comprising:

when the computed hash value and the trusted hash value are indicative of a same state of a computer system, providing security data from a trusted source to an application in execution on the system.

21. (Previously Presented) The method of claim 20, further comprising:

when the computed hash value and the trusted hash value are other than indicative of a same state of the system, sending a notification.

22. (Currently Amended) The method of claim ~~1~~79, wherein the second data includes DLL tables.

23. (Cancelled)

24. (Currently Amended) The method of claim ~~1~~79, wherein the predetermined data is hashed in an absolute memory location independent fashion.

25. (Currently Amended) The method of claim ~~1~~79, wherein if the computed hash value and the trusted hash value substantially match, there is no unauthorized executable code in the computer system.

26. (Currently Amended) The method of claim ~~1~~79, further comprising performing a user authorization process for verifying that a user is authorized.

27. (Previously Presented) The method of claim 26, wherein the one or more applications executing in the computer system includes at least one untrusted application and at least one trusted application, and further comprising transmitting a password request from the at least one untrusted application to the at least one trusted application.

28. (Previously Presented) The method of claim 27, wherein the transmitting a password request from the at least one untrusted application to the at least one trusted application is in response to a user's attempt to access a data file associated with the at least one untrusted application.

29. (Previously Presented) The method of claim 27, wherein the user authorization process comprises:

detecting the password request from the at least one untrusted application by the at least one trusted application;

prompting the user to input authorization information; and

comparing the input authorization information with information retrieved from the at least one trusted application, wherein if the input authorization information successfully compares with the information retrieved from the at least one trusted application, the user is an authorized user.

30. (Previously Presented) The method of claim 29, wherein the hashing data, retrieving a trusted hash value and the comparing the computed hash value with the trusted hash value are carried out if the input authorization information successfully compares with the information retrieved from the at least one trusted application.

31. (Previously Presented) The method of claim 29, wherein the input authorization information comprises at least biometric information, and wherein the comparing includes comparing the at least biometric information with a previously stored biometric template.

32. (Previously Presented) The method of claim 29, wherein the at least one trusted application includes a user verification database, and wherein the input authorization information is compared with information retrieved from the user verification database.

33. (Currently Amended) The method of claim ~~1~~79, wherein the at least one trusted application includes a hash generator and wherein the hashing data is carried out in the hash generator.

34. (Currently Amended) The method of claim ~~1~~79, wherein the trusted hash value is encrypted.

35. (Previously Presented) The method of claim 34, wherein the trusted hash value is digitally signed.

36. (Previously Presented) The method of claim 34, further comprising:

decrypting the encrypted trusted hash value; and

comparing the decrypted trusted hash value with the computed hash value, wherein if the computed hash value and the decrypted trusted hash value

substantially match, there is no unauthorized executable code in the computer system.

37. (Previously Presented) The method of claim 35, further comprising:

decrypting the digitally-signed trusted hash value; and

comparing the decrypted trusted hash value with the computed hash value wherein if the computed hash value and the decrypted trusted hash value substantially match, there is no unauthorized executable code in the computer system.

38. (Previously Presented) The method of claim 27, wherein the trusted hash value is digitally signed and further comprising:

decrypting the digitally-signed trusted hash value;

comparing the decrypted trusted hash value with the computed hash value; and

refusing the password request from the at least one untrusted application if the computed hash value and the decrypted trusted hash value do not substantially match.

39. (Currently Amended) The method of claim ~~1~~79, wherein the data storage comprises at least a volatile memory.

40. (Currently Amended) The method of claim ~~1~~79, wherein the data storage comprises at least a disk drive.

41. (Currently Amended) The method of claim ~~1~~79, further comprising:

determining that the computer system is in a known state;

hashing data representing the known state of the at least the one application executing in the computer system using the selected hashing process to create the trusted hash value; and

encrypting the trusted hash value.

42. (Previously Presented) The method of claim 41, further comprising:

retrievably storing the trusted hash value in the data storage.

43. (Previously Presented) The method of claim 41, wherein the determining comprises:

performing a user authorization process to determine an authorized user; and

receiving a command from the authorized user that the computer system is in a known state.

44. (Previously Presented) The method of claim 41, wherein the trusted hash value is digitally signed.

45. (Previously Presented) The method of claim 34, wherein the computer system includes a plurality of networked computers, and wherein the encrypted trusted hash value is stored in a secure one of said plurality of computers, the method further comprising:

receiving in the secure computer, the computed hash value transmitted from at least a first computer;

decrypting the encrypted trusted hash value in the secure computer;

wherein the comparing of the decrypted trusted hash value with the computed hash value occurs in the secure computer; and

if the computed hash value and the trusted hash value substantially match—

retrieving a password from a memory in the secure computer; and

transmitting the retrieved password to the at least a first computer.

46. (Previously Presented) The method of claim 45, wherein the one or more applications executing in the computer system includes at least one untrusted application executing on the at least a first computer and at least one trusted application executing on the at least a first computer, the method further comprising:

detecting a password request from the at least one untrusted application by the at least one trusted application;

prompting the user to input authorization information;

comparing the input authorization information with information retrieved from the at least one trusted application; and

wherein if the input authorization information successfully compares with the information retrieved from the at least one trusted application, the user is an authorized user.

47. (Previously Presented) The method of claim 45, wherein the trusted hash value is digitally signed, and further comprising:

decrypting the digitally-signed trusted hash value in the secure computer; and

comparing the decrypted trusted hash value with the computed hash value in the secure computer, wherein if the computed hash value and the decrypted trusted hash value substantially match, there is no unauthorized executable code in the at least a first computer.

48. (Previously Presented) The method of claim 47, further comprising:

refusing the password request from the at least one untrusted application if the computed hash value and the decrypted trusted hash value do not substantially match.

49. (Previously Presented) The method of claim 45, further comprising:

determining that the at least a first computer is in a known state;

hashing data representing the known state of the at least one application executing in the at least a first computer using the selected hashing process to create the trusted hash value; and

encrypting the trusted hash value.

50. (Previously Presented) The method of claim 49, further comprising:

transmitting the encrypted trusted hash value to the secure computer; and

storing the encrypted trusted hash value in the secure computer.

51. (Previously Presented) The method of claim 45, further comprising:

encrypting the computed hash value in the at least a first computer prior to transmission; and

decrypting the computer hash value in the secure computer.

52. (Previously Presented) The method of claim 45, wherein the retrieved password is encrypted, and further comprising:

decrypting the retrieved password in the at least a first computer.

53. (Previously Presented) The method of claim 45, further comprising:

transmitting an incorrect password to the at least one untrusted application of the at least first computer if the computed hash value and the trusted hash value do not substantially match.

54. (Currently Amended) The method of claim 45, further comprising:

transmitting a lock command to the at least one untrusted application of the at least first computer if the computed hash value and the trusted hash value do not substantially match.

55. (Currently Amended) The method of claim ~~2~~79, further comprising:

prompting a user to verify that the unauthorized executable code is from a known source if the computed hash value and the trusted hash value do not substantially match.

56. (Previously Presented) The method of claim 45, wherein if the computed hash value and the trusted hash value do not substantially match, there is unauthorized executable code in the at least a first computer, and further comprising:

prompting a user to verify that the unauthorized executable code is from a known source.

57. (Cancelled)

58. (Currently Amended) The method of claim 79~~1~~, wherein the known state is an initial state of an operating system within the computer system.

59. (Cancelled)

60. (Previously Presented) The method of claim 45, wherein the known state is an initial state of an operating system within the computer system.

61. (Currently Amended) A system for detecting unauthorized executable resident in a computer system, the system comprising a computer processor programmed to perform the method comprising:

receiving user authorization information;

authenticating the user authorization information to perform at least one of authorize and identify a user;

when the user is at least one of authorized or identified, requesting security data of the user;

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a ~~known~~ secure state of the one or more applications executing in the computer system, wherein the second data includes data from at least a system memory location indicative of the at least one application executing within the computer system; and

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system.

62. (Cancelled)

63. (Currently Amended) A computer readable storage medium for detecting unauthorized executable code resident in a computer system, the computer readable storage medium having stored thereon instructions that, when executed, perform a method comprising:~~The computer readable storage medium of claim 62, further comprising computer executable instructions for~~

receiving user authorization information;

authenticating the user authorization information to perform at least one of authorize and identify a user; ~~and~~

when the user is at least one of authorized or identified, requesting security data of the user~~.~~;

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a secure state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system; and

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system.

64. (Previously Presented) The computer readable storage medium of claim 63, wherein the authorization data is at least a biometric information sample, and wherein the authenticating includes comparing the at least a biometric information sample to a previously stored biometric template.

65. (Previously Presented) The computer readable storage medium of claim 63, further comprising computer executable instructions for, when the comparison is indicative of other than unauthorized executable code resident in a computer system, providing the requested security data relating to the user.

66. (Currently Amended) The computer readable storage medium of claim ~~62~~63, further comprising computer executable instructions for receiving a request for security data from an application in execution in the computer system, and when the comparison is indicative of other than an unauthorized executable program resident in a computer system, providing security data to the application.

67. (Currently Amended) The computer readable storage medium of claim ~~62~~63, wherein the trusted hash value and the computed hash value are determined by a same trusted security application executing locally on a processor of a same computer system at different times, the trusted hash value determined when the known state is a secure state.

68. (Previously Presented) The computer readable storage medium of claim 67, wherein the trusted hash value is digitally signed.

69. (Previously Presented) The computer readable storage medium of claim 68, wherein the hashing data includes verifying an authenticity of the digitally signed trusted hash value.

70. (Currently Amended) The computer readable storage medium of claim ~~62~~63, further comprising computer executable instructions for receiving a request for security data from an application in execution in the computer system, and, when the authenticity of the digitally signed trusted hash value is verified and the comparison is indicative of other than unauthorized executable code resident in a computer system, providing the requested security data to the application.

71. (Previously Presented) The computer readable storage medium of claim 70, wherein the application and the predetermined hashing process are both executed on a same processor of the computer system.

72. (Previously Presented) The computer readable storage medium of claim 69, further comprising computer executable instructions for when the computed hash value and the trusted hash value are other than indicative of a secure state, issuing a notification that unauthorized executable code is detected within the computer system.

73. (Previously Presented) The computer readable storage medium of claim 72, further comprising computer executable instructions for, when the computed hash value and the trusted hash value are other than indicative of a secure state, preventing access to the computer system.

74. (Previously Presented) The computer readable storage medium of claim 68, further comprising computer executable instructions for transmitting the trusted hash value to a second other computer system in communication with the computer system and retrievably storing the computed hash value within the second other computer system.

75. (Previously Presented) The computer readable storage medium of claim 74, further comprising transmitting the computed hash value to the second other computer system for comparison with the trusted hash value by a processor of the second other computer system.

76. (Previously Presented) The computer readable storage medium of claim 75, wherein the computed hash value is determined in dependence upon the predetermined data existing in memory within the computer system and some time dependent data of the computer system.

77. (Previously Presented) The computer readable storage medium of claim 76, wherein the second other computer system includes a trusted source and wherein security data is stored for provision to applications in execution on systems that are known to be secure.

78. (Currently Amended) A system for detecting unauthorized executable resident in a computer system, the system comprising:

means for receiving user authorization information;

means for authenticating the user authorization information to perform at least one of authorize and identify a user;

means for requesting security data of the user when the user is at least one of authorized or identified;

means for hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value,

wherein the first data includes data representing a current state of at least one application executing within the computer system;

means for retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a ~~known~~ secure state of the one or more applications executing in the computer system, wherein the second data includes data from at least a system memory location indicative of the at least one application executing within the computer system; and

means for comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system.

79. (New) A method of detecting unauthorized executable code resident in a computer system, the method comprising:

receiving user authorization information;

authenticating the user authorization information to perform at least one of authorize and identify a user;

when the user is at least one of authorized or identified, requesting security data of the user;

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a secure state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system; and

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system.

80. (New) A method of detecting unauthorized executable code resident in a computer system, the method comprising:

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a secure state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system;

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system; and

performing a user authorization process for verifying that a user is authorized,

wherein the one or more applications executing in the computer system includes at least one untrusted application and at least one trusted application, and wherein the method further comprises transmitting a password request from the at least one untrusted application to the at least one trusted application, and

wherein the trusted hash value is digitally signed, and the method further comprises—

decrypting the digitally-signed trusted hash value;

comparing the decrypted trusted hash value with the computed hash value; and

refusing the password request from the at least one untrusted application if the computed hash value and the decrypted trusted hash value do not substantially match.

81. (New) A method of detecting unauthorized executable code resident in a computer system, the method comprising:

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a secure state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system, and further wherein the trusted hash value is encrypted;

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system,

wherein the computer system includes a plurality of networked computers, and wherein the encrypted trusted hash value is stored in a secure computer of said plurality of computers, the method further comprising—

receiving in the secure computer, the computed hash value transmitted from at least a first computer; and

decrypting the encrypted trusted hash value in the secure computer,

wherein comparing the computed hash value with the decrypted trusted hash value occurs in the secure computer;

if the computed hash value and the trusted hash value substantially match—

retrieving a password from a memory in the secure computer; and

transmitting the retrieved password to the at least a first computer; and

if the computed hash value and the trusted hash value do not substantially match—

transmitting an incorrect password and/or a lock command to the at least one untrusted application of the at least first computer.

82. (New) A method of detecting unauthorized executable code resident in a computer system, the method comprising:

hashing first data stored in data storage within the computer system using a selected hashing process to determine a computed hash value, wherein the first data includes data representing a current state of at least one application executing within the computer system;

retrieving a trusted hash value, wherein the trusted hash value was created using the selected hashing process applied to second data representing a secure state of the one or more applications executing in the computer system, wherein the second data includes a system memory location indicative of the at least one application executing within the computer system, and further wherein the trusted hash value is encrypted;

comparing the computed hash value with the trusted hash value to determine whether there is unauthorized executable code in the computer system,

wherein the computer system includes a plurality of networked computers, and wherein the encrypted trusted hash value is stored in a secure one of said plurality of computers, and the method further comprises—

receiving in the secure computer, the computed hash value transmitted from at least a first computer;

decrypting the encrypted trusted hash value in the secure computer,

wherein comparing the computed hash value with the decrypted trusted hash value occurs in the secure computer;

if the computed hash value and the trusted hash value substantially match—

retrieving a password from a memory in the secure computer; and

transmitting the retrieved password to the at least first computer; and

if the computed hash value and the trusted hash value do not substantially match—

prompting a user to verify that any unauthorized executable code in the at least first computer is from a known source.